Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended) A method comprising:

determining whether execution of an instruction of a first thread executed in a processor potentially causes a long latency; and

switching to a second thread based on the determination, ~~and~~ executing one or more instructions of the second thread in the processor, and storing a result of the one or more instructions of the second thread in the processor.

Claim 2 (original) The method of claim 1, further comprising executing at least one additional instruction in the first thread while preparing to switch to the second thread.

Claim 3 (currently amended) The method of claim 1, wherein the determining is based on a stochastic ~~analysis~~ modeling of whether the instruction will result in a long latency.

Claim 4 (currently amended) The method of claim 1, wherein the determining comprises applying the instruction to a lookup table in [[a]] an instruction decoder of the processor ~~pipeline~~, the lookup table including entries corresponding to predetermined conditions.

Claim 5 (currently amended) The method of claim 4, further comprising providing a feedback signal from [[an]] the instruction decoder to an instruction fetch unit to switch to the second thread if the instruction matches an entry in the lookup table.

Claim 6 (original) The method of claim 1, wherein the long latency comprises less than ten processor cycles.

Claim 7 (canceled)

Claim 8 (currently amended) A method comprising:

switching from a first thread to a second thread if a condition that potentially results in a stall of a processor pipeline occurs during execution of the first thread in the processor pipeline, and executing at least one instruction of the second thread to reduce or prevent the stall.

Claim 9 (currently amended) The method of claim 8, further comprising determining whether the condition occurs by comparing an instruction to entries in a lookup table of an instruction decoder of the processor pipeline.

Claim 10 (previously presented) The method of claim 8, further comprising executing at least one additional instruction in the first thread after the condition occurs and before switching to the second thread.

Claim 11 (original) The method of claim 8, wherein the condition is based on a stochastic model.

Claim 12 (currently amended) The method of claim 8, further comprising providing a feedback signal ~~from~~ generated in an instruction decoder to an instruction fetch unit to switch to the second thread.

Claim 13 (currently amended) ~~An article comprising a machine~~ A computer-readable ~~storage~~ medium ~~containing~~ storing instructions that [[if]] are executed to enable a system to:

switch from a first thread to a second thread if a condition that potentially results in a stall of a processor pipeline occurs during execution of the first thread in the processor pipeline, and execute at least one instruction of the second thread.

Claim 14 (currently amended) The ~~article~~ computer-readable medium of claim 13, further comprising instructions that [[if]] are executed to enable the system to determine whether the condition occurs by comparing an instruction to entries in a lookup table of an instruction decoder of the processor pipeline, and provide a feedback signal from the instruction decoder to an instruction fetch unit to switch to the second thread if the instruction matches an entry in the lookup table.

Claim 15 (currently amended) The ~~article~~ computer-readable medium of claim 13, further comprising instructions that [[if]] are executed to enable the system to execute at least one additional instruction in the first thread while the system prepares to switch to the second thread.

Claim 16 (currently amended) The ~~article~~ computer-readable medium of claim 13, further comprising instructions that [[if]] are executed to enable the system to send a feedback signal generated in an instruction decoder to an instruction fetch unit to cause the switch from the first thread to the second thread.

Claim 17 (currently amended) An apparatus comprising:

a processor pipeline having a feedback loop coupled between a first pipeline stage and a second pipeline stage to provide a feedback signal ~~from~~ generated in the second pipeline stage back to the first pipeline stage to cause the processor pipeline to switch from a first thread to a second thread, the feedback signal to originate from a location in the processor pipeline before instruction execution.

Claim 18 (original) The apparatus of claim 17, wherein the feedback signal is coupled between an instruction decoder and an instruction fetch unit.

Claim 19 (original) The apparatus of claim 18, wherein the instruction decoder is coupled to provide the feedback signal to the instruction fetch unit when a predetermined condition occurs.

Claim 20 (original) The apparatus of claim 19, wherein the instruction decoder includes logic to determine when the predetermined condition occurs.

Claim 21 (previously presented) The apparatus of claim 19, wherein the instruction decoder includes a lookup table to store a list of predetermined conditions to which an instruction is compared.

Claim 22 (currently amended) A system comprising:

a processor pipeline including an instruction decoder and an instruction fetch unit, the processor pipeline having a feedback loop coupled to provide a feedback signal ~~from an~~ generated in the instruction decoder to [[an]] the instruction fetch unit to cause the processor pipeline to switch from a first thread to a second thread, the feedback signal to originate from a location in the processor pipeline before instruction execution; and

a wireless interface coupled to the processor pipeline via a bus.

Claim 23 (original) The system of claim 22, further comprising at least one storage device to store code to enable the processor pipeline to switch from the first thread to the second thread if a predetermined condition occurs during execution of the first thread.

Claim 24 (original) The system of claim 23, wherein the at least one storage device includes code to enable the processor pipeline to execute at least one additional instruction in the first thread while the system prepares to switch to the second thread.

Claim 25 (cancel)

Claim 26 (currently amended) The system of claim [[25]] 22, wherein the instruction decoder is coupled to provide the feedback signal to the instruction fetch unit when a predetermined condition occurs.

Claim 27 (original) The system of claim 26, wherein the instruction decoder includes logic to determine when the predetermined condition occurs.

Claim 28 (currently amended) The system of claim 26, wherein the instruction decoder includes a lookup table that includes a list of predetermined conditions to which an instruction is applied, the instruction decoder to provide the feedback signal to the instruction fetch unit if the instruction matches an entry in the lookup table.

Claim 29 (original) The system of claim 22, wherein the wireless interface comprises a dipole antenna.

Claim 30 (previously presented) The method of claim 4, wherein the predetermined conditions comprise instruction types.

Claim 31 (new): The method of claim 9, further comprising sending a feedback signal from the instruction decoder to an instruction fetch unit if the instruction matches an entry in the lookup table.